MySQL Task

Task 1: Database and Table Creation

Objective: Create a database and a few tables.

1. Create a database named `company_db`.
2. Create a table named `employees` with the following structure:
   - `id` (INT, Primary Key, Auto Increment)
   - `name` (VARCHAR(100), NOT NULL)
   - `position` (VARCHAR(50))
   - `salary` (DECIMAL(10,2))
   - `joining_date` (DATE)

Task 2: Data Insertion

Objective: Insert records into the `employees` table.

1. Insert at least five employee records with different positions and salaries.
2. Ensure some employees have the same positions for query testing.

Task 3: Basic Queries

Objective: Retrieve data using SQL queries.

1. Select all employees.
2. Retrieve employees with a salary greater than 50,000.
3. Retrieve employees who joined after `2020-01-01`.

Task 4: Updating and Deleting Data

Objective: Perform UPDATE and DELETE operations.

1. Update the salary of an employee with `id = 3` to `70,000`.
2. Delete an employee with `id = 5`.

Task 5: Joins and Relationships

Objective: Create and query related tables.

1. Create a `departments` table with:
   - `id` (INT, Primary Key, Auto Increment)
   - `department_name` (VARCHAR(50))
2. Add a `department_id` column to `employees` and establish a foreign key.
3. Retrieve employee names along with their department names using `JOIN`.

Task 6: Aggregate Functions and Grouping

Objective: Utilize aggregate functions for analysis.

1. Find the average salary of employees.
2. Count the number of employees in each position.
3. Get the highest salary in each department.

Task 7: Indexing and Performance Optimization

Objective: Improve query performance.

1. Create an index on the `name` column of `employees`.
2. Explain how indexing helps in optimizing queries.

Task 8: Stored Procedures and Triggers

Objective: Implement stored procedures and triggers.

1. Write a stored procedure to get employees with salaries above a given threshold.

2. Create a trigger that updates a log table whenever an employee’s salary is updated.

Task 9: Backup and Restore

Objective: Secure data using backup and restore methods.

1. Take a backup of the `company_db` database.
2. Restore the database from the backup.
